The relationship between Media and Public Interest

IT-University of Copenhagen
Data in the Wild (KSDWWVD1KU)
Wednesday 13th of December 2023

Steinar Slette Marek Gala Manuel Knepper Niclas Claßen

***Abstract*—The discourse of topics in the media is largely driven by big news sites. At the same time, a big part of the population looks to search engines to get information about current events. We investigated the media coverage of specific categories of topics in two major news sites based in the United Kingdom - *The Guardian* and *Sky News* - in two specific timeframes, and compared the results to their respective google search popularity for the United Kingdom. We use web scraping and API requests to gather the data, along with several other techniques for data processing. We use both manual and automatic annotation to classify each of the articles with a category. In our analysis, we show that the topics that news sites report on change between different timeframes. We also detect a low correlation between media coverage and public interest, even though we are unable to attest causation.**

***Index Terms*—Scraping, API, Google Trends, Media coverage**

I. Introduction

In contemporary times, large news sites are one of the most frequented sources for information. According to research on news consumption, online sources are the second most used platforms for news behind broadcast TV. Online sources are used by 68% of UK adults, and 83% of young adults aged 16-24 [1]. We postulate that this popularity can lead to news sites influencing the discourse of topics in public in general.

Data about published articles from news sites is largely not archived very robustly. Data can be retrieved by accessing official APIs provided by the news sites, though these are rather scarce. As there didn't seem to be any other reasonable method to get the article data from the news websites, we resorted to using scraping methods.

Search engines like Google also aggregate data about search popularity of the queries the Google Users search for. Google provides a popularity ranking on what search terms were particularly searched for, broken down by date and region. This data can be accessed by scraping, however there are also libraries that provide unofficial APIs to Google Trends.

We propose a dataset to investigate the relationship between the topics of articles that are published by news providers and their search popularity made by Google Users in the United Kingdom. The intended use of the proposed dataset is to investigate if the topics that traditional news outlets report on coincide with the topics trending on Google Trends. Optionally, the dataset could be used to analyze differences in the political affiliation of news sources (left/center/right) and the impact it has on the distribution of published topics. Another possible use of the dataset could be in natural language processing on article texts from different news sources, for example to investigate how different news platforms report on the same issues.

To be able to draw comparisons between different timeframes, we gathered data for January 2022 and January 2023.

A. Research question

Is there a correlation between media coverage in particular categories of topics and the interest of the population in the given categories?

- Can we observe a shift in topics that get reported on from 2022 to 2023?
- Do we see an increase in Google Trends for a topic after the topic has been represented in the media? If so, to what degree do we see a change in public interest?
- Alternatively, does the public interest influence media representation instead of media influencing public interest?

We hypothesize that there will be some correlation between media coverage and public interest for specific topics across the time period we are looking at. A possible limitation in accurately answering this question is the amount of data we have available. For example, we focus on some of the biggest providers of news in the UK for this analysis, but there are still many other sources that may influence the interest of the population. Therefore, the results may not reflect the entire reality of the effect media has on the public interest.

B. Selection of News Sites

The selection of the news sites specified in this report is based on several attributes. Firstly, we focused on the average number of visitors each site had. We derived those statistics from the online outlet PressGazette [2]. Secondly, given that the news site had a sufficient amount of readers to enable

further analysis, is the data in question readily available either through an API or possible to gain by implementing some scraping procedure? We checked the top 20 news sites mentioned in [2] for those properties, with the goal of choosing a selection for further analysis.

Considering these specifications we ended up choosing the news providers *Sky News*, *BBC* and *The Guardian* as our target news providers. *The Guardian* is one of the few news sites that provide a clean programmatic access to their archive. For the other two, we resorted to scraping the data with the assistance of the Wayback Machine maintained by the Internet Archive [3].

C. Delimitations

Specifying the geographical region that we want to focus on helps us with selecting news sources for analysis, without having to account for regional differences and biases in reporting style. As such, we have limited the target region to the United Kingdom, considering that this region consists of many large news providers that publish articles written in English, thus it provides a sufficient sample for answering our research question.

II. Data collection

The focus of this study is on the process of collecting and processing data. The data gathering process is a combination of several methods. Considering that we wanted to gain insight into articles published in certain timeframes for several news providers, the use of different techniques was a necessity. We choose to gather data over two different timeframes, namely January 2022 and January 2023, to be able to compare differences between them and to see how the focus of reporting might have shifted in between those timeframes. For January 2022 we collected the data of 6.181 articles from *The Guardian*, and 1.929 articles from *Sky News*, while for January 2023 we gathered 5.909 articles from *The Guardian* and 2.505 articles from *Sky News* respectively.

A. News articles

For *The Guardian* we used an official API that allowed us to get the necessary data. The news providers *Sky News* and *BBC* did not provide an official API nor an archive that would allow us to scrape the news site directly for a specific timeframe. For that reason we devised a scraping procedure that would run on snapshots provided by the Wayback Machine for the news pages for the given dates.

We are only interested in certain data for every article: the title, the article text, the publishing date, the author(s), tags that are associated with the article and the URL under which the article can be found.

1) The Guardian: The API to access the archive of articles by *The Guardian* is based on the REST architectural style. To gain access to the API, we need to register for a free developer API key, which is limited to one API call per second and 500 API calls per day. This is sufficient for our project, if these limits prove to be too restrictive there is also the option of a paid commercial API key [4].

We send a request to the API with the *requests* package in Python. To get the response that we are looking for we use the following parameters for our API call: a filter on article to exclude videos and blog posts from the response, and a filter to only get tags that are either of the type ”contributor” indicating an author of the article, or of the type ”keyword” which describe the content of the article. We also include the whole body of the article to extract the article text [5].

When we get the response from the API, which is provided to us in JSON format, we parse through every page using the *json* package in Python. The response of the API includes a maximum of 50 articles per results page, we use pagination with the ”page” parameter provided by the API to access all results pages. We collect all the information that we outlined above by accessing the response like a nested object. If certain data for an article is missing, e.g. authors of the article, we fill in the missing value as ”N/A”.

Once we collected all the data for one article, the data gets appended to a list of list and we move on to the next article. This way we parse through all results pages. Once parsing is complete, we convert the list to a pandas Data Frame which we then convert to a csv file for further analysis.

2) Sky News: To scrape the news articles from *Sky News* we utilized the Python libraries *Newspaper3k* [6] and *Beautiful Soup* [7]. *Newspaper3k* as the main component of our scraper is built on top of the libraries *requests* [8] and *lxml* [9]. However, before we can scrape the articles we have to collect the relevant URLs of captures made by the Wayback Machine. Therefore we made a request with a GET query where we included the URL and timeframe of interest to the Wayback CDX server. We extended the timeframe by one day, to cover the edge case where an article got published after the last capture on the last day for the requested time period. The Wayback CDX server then returns the result as a JSON array from which we use the ‘timestamp’ and ‘original’ column to put together a final Wayback Machine URL for each capture. For January 2022 and January 2023 we ended up with 1.275 and 3.078 URLs of captures.

The *Newspaper3k* library is used to parse the news articles by accessing the final URLs of the captures of the Wayback Machine. We then make use of the functionalities of *Beautiful Soup* to clean the parsed data and filter out relevant fields like the publishing date of an article which is not automatically detected by *Newspaper3k*. If we encounter a missing value

for a field, we fill in the missing value as ”N/A”. To not parse an article again from another capture we keep track of the specific URL of an article as it can be found under *Sky News* and only include articles that we have not processed so far.

The dataset produced by this procedure contains the columns ”Title”, ”Date”, ”Authors”, ”Tags”, ”Text” and ”Url”. The data contained in the ”Authors” and ”Text” columns is not presently utilized in this study, but is still included as a potential source for further analysis.

One challenge that occurred on the way of implementing the scraping process based on *Newspaper3k* library is that the base URL for the URLs of captures we wanted to scrape always stays the same. Therefore the request method in *Newspaper3k* interprets this as wanting to scrape the same page over and over again even though the subdomain changes. Instead of scraping the new results, it returns the already scraped results from the cache locally stored on the local machine. For this reason we have to delete the cache which is stored on the local machine after each scrape.

3) BBC: We intended to involve *BBC* in our analysis. However, the layout of a large number of articles limited our scraping capabilities. For this reason, we were unable to perform a sufficiently large scrape for *BBC*. Considering the fact that the results from the scrape of *BBC* could potentially introduce bias due to the low volume of successfully scraped data, we decided to eliminate the articles affiliated with *BBC* from the analysis. The dataset containing the scraped articles from *BBC* can be found in the *raw* data folder, but is not included in the processed dataset.

The scraping procedure we created for *BBC* was relatively similar to the procedure created for *Sky News*, we made use of the Python libraries *Newspaper3k* and *Beautiful Soup* and gained the results from snapshots present on the Wayback Machine. More information about the following operation was detailed in the *Sky News* section. The issue detected with scraping data from the *BBC* news pages is based on the structure of the majority of the articles. The structure switches between a standard HTML layout that can easily be scraped, and dynamic infinite scroll-lists. In the given timeframe, we were not successful in implementing a scraping procedure for the dynamically generated article web pages. This is because the web page continuously updates upon scrolling, revealing another article after having scrolled through the current one. There may very well exist a solution to this problem, but we decided to put it outside of the scope of this project. The interested reader is referred to solutions such as *Selenium* [10] that are more optimized towards scraping dynamic web pages.

B. Google Trends

While Google Trends does not provide an official API, there is a way to get access to Google Trends without setting up a dedicated crawler. The ”unofficial API for Google Trends” library *pytrends* [11] for Python allowed us to get the relative popularity for all the tags we have extracted from the collected articles. The only data we deemed appropriate for use in the analysis were retrieved by the API method *interest over time*. The request returns historical relative popularity for the tags, same as would be seen on the Google Trends' Interest Over Time section.

Due to limitations of the API, the request may contain 5 tags at most. To get around this limitation and retrieve all the results, we had to split our request in batches of size five or less tags. Conveniently for us, the results within one batch are relative to each other. The relative popularity is always within range 0-100. If the popularity is 0, there is not enough data to decide the popularity of the tag. If the popularity is 100, it is the most popular tag within the batch for the day. To be able to combine and normalize the results for all the batches, we searched for a tag that is somewhere in the middle of the popularity range, and used it to normalize the results.

While working on this project, we encountered some issues in extracting the data using the *pytrends* API. The most common error we received was 429, which indicates the user has sent too many requests. It might be possible that Google recognizes certain IP addresses as suspicious and does not return any data after suspicious requests have been detected. Google frequently make changes to their services, and these changes are frequently preventing the data from being scraped. After reviewing the *pytrends* GitHub repository, we found an issue [12] with numerous solutions. It seemed that different solutions worked for different people. We managed to reduce the error rate by increasing the number of retries and changing the request method from GET to POST, as suggested by a Stack Overflow article [13].

III. Data Processing

The data processing constitutes one of the larger processes in this survey. This section describes all the steps taken in order to ensure that the data is readily available for further analysis and can be used as is, not only by us, but also by subsequent researchers without further processing of the data.

A. Pre-Processing

For some data, pre-processing was done in the data gathering phase. Some general data cleaning could be implemented in the scraping procedures, like converting strings to lowercase to ensure that the format was similar across the various datasets. Some general processing also needed to be done in order to accommodate the different formats of the data on the various websites. Superfluous columns that would not be needed in analysis were dropped during the pre-processing stage, and parsing was done on dates to ensure the format was the same across the sources. The raw data, before the pre-processing is applied to it, is preserved on the GitHub repository in the *raw* folder, so that

it may also be used for different purposes.

Additionally, we had to normalize the data from Google Trends, so that the individual trend scores were on the same scale and could be compared as a whole. In order to do that, we combined two approaches.

In the first approach, we send five batches. We choose one tag at random, and include it in each batch - that is the predefined tag. We take the average of the time series for each tag within a batch, and take median value of the average for each of the responses. In the second approach, we send five batches that each contain five independent tags. Therefore we send five batches that each contain independent elements, and we take the average and median again. We then combine the results from the two approaches, and finally select the median from the ten tags. This will be our predefined tag that we will use to put the results from the different batches on a common scale. The predefined tag will be involved in each batch for that reason. This approach loosely follows suggestions in a Medium article [14]. Applying the first method maintains the consistency of relative scores, but the predefined tag for the batches might be on either extreme of the popularity range, which would skew the results. To reduce the risk of picking a biased predefined tag, we also applied the second method.

After collecting the data for all the tags, we normalize the relative popularity by dividing the popularity of each tag inside a batch by the popularity of the predefined tag. We do this for each day in the time series. If the popularity of the predefined keyword is zero, we set it to one - meaning it will have no change on the results. In the table I, you can observe how normalized results may appear for two days in January 2022.

TABLE I
NORMALIZED RESULTS EXAMPLE

Date	Tag 1	Tag 2	Tag 3	Tag 4	Predefined Tag
2022.01.01	45.7	55.5	27.8	11.4	1
2022.01.02	47.1	52.3	13	15.7	1

Note how the predefined tag value is one in all the rows, as we have divided the entire row by its value. We propose that now, the predefined tag is used to put all the tags on the same scale, while acknowledging a margin for error. If the popularity value of a tag is high, for example 80 we assume that it is considerably more popular than the average tag. The dataset containing the Google Trends data was originally structured with each tag as a column and the relevant dates and score for each date as row values, as shown in the table above. Besides normalizing the results, we also transposed the Data Frame in order to improve readability.

To uncover whether there exists a connection between media coverage and public interest, we had to append a new column in the articles dataset named *score*. It is calculated by getting all tags related to an article and taking an average of their represented scores in the *pytrends* normalized dataset for the date that the article was published. In this way we get an estimated popularity score for each article based on the attached tags which can be utilized to visualize the overall popularity of each category and its connection to the frequency of articles published in the category.

B. Annotation

The main bulk of the annotation was done by utilizing the GPT-3.5-Turbo API [15] asking it to classify each article based on the article headline and the tags. To verify that the results we got from the API are sufficiently accurate, we manually annotated two sample datasets with / 800 articles. We also conduced a test to compare the accuracy of the manual and GPT annotation. The main objective of the annotation was to classify which category of topics an article referred to. We decided on using these categories for classification, as they are often also present on news sites:

- Politics
- Business and Economy
- Environment
- Sports
- Entertainment and Culture
- Science and Technology
- Health

1) Manual Annotation: The manual annotation was performed by two individuals to avoid introducing bias in the annotation. Each annotator annotated two datasets - one per year. Same as with the automatic annotation, only the title of each article and the associated tags were used.

2) Automatic Annotation: We utilized the OpenAI API to access the GPT-3.5-Turbo model for automating the annotation process for the articles and their affiliation to a specific category. In general, the API works by receiving a prompt specifying the task and an input the task is to be performed on. In our case, the input consists of the headline and the tags of an article. Based on that, the GPT-3.5-Turbo classifies the article as belonging to a singular category.

Because of the API limitations, we first tried to annotate the articles in batches. The prompt used for this can be found under VII-A. This worked well for most cases, but for some it was not able to annotate all of the instances or returned other than the predefined categories.

Consequently, we changed the strategy by making a separate request for each article and numbering the predefined categories. We found that this resolved the issues we were experiencing. The prompt used for this can be found under VII-B.

We also annotated the articles without tags. However, these are not used in the final analysis due to the fact that we cannot relate them to the *pytrends* data by calculating the score column.

3) Analysis of Annotation: To verify that the annotation performed by GPT-3.5-Turbo was sufficiently accurate we analyzed the accuracy between manual annotation and the annotation performed by GPT-3.5-Turbo.

TABLE II
ANNOTATION ACCURACY IN %

Annotation Accuracy in %			
Year	Case 1	Case 2	Case 3
2022	82.78	85.24	83.81
2023	80.10	81.23	80.03

Table II describes the accuracy of the annotation for three different cases. Case 1, the accuracy of the annotation between two human annotators. Case 2, the accuracy of the annotation between two human annotators and GPT-3.5-Turbo on the subset of articles where the two human annotators agree, and Case 3, the accuracy where GPT-3.5-Turbo agrees with at least one human annotator.

The resulting accuracy is acceptable for further analysis, we see that the accuracy of the annotation performed by GPT-3.5-Turbo when compared to a human annotator is comparable to the accuracy between two human annotators. This is expected due to there being some articles that arguably could be labeled as belonging to multiple categories.

We decided on using the dataset created with the automatic annotation for further analysis.

IV. ANALYSIS

To see which categories of topics were most represented in the media outlets we chose, and also if those representations change from 2022 to 2023, we used the annotations made by GPT-3.5-Turbo. The results are displayed in Fig. 1.

One clear trend than can be observed is the decreased reporting on topics in the category ”Health”, while articles that make up that category account for 13.54% of the total amount of articles in 2022, only 9.61% of articles published in 2023 are classified as articles belonging to that category. Our hypothesis is that this is linked to the Covid-19 pandemic, which was more acute in the public focus in 2022 rather than 2023. When looking at the frequency of specific tags for articles in 2022 and 2023 we see that the tag ”Coronavirus” is used 1.155 times in 2022 which makes it the second most frequent tag that year, and only 99 times in 2023, making it the 52nd most frequent tag that year, as depicted with normalized values in Fig. 2.

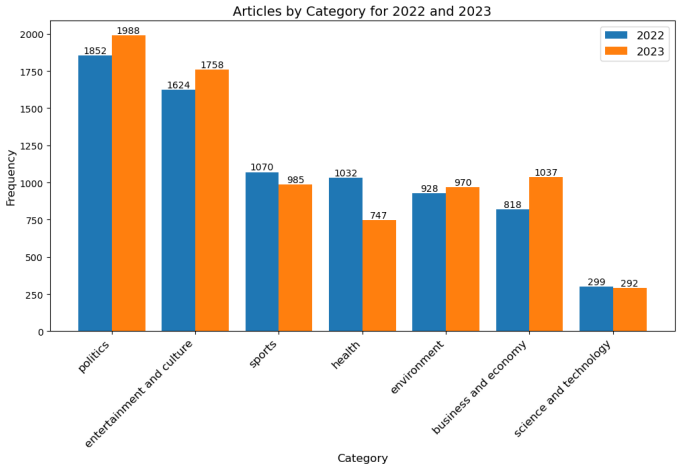

Fig. 1. Number of articles published in each category both for 2022 and 2023.

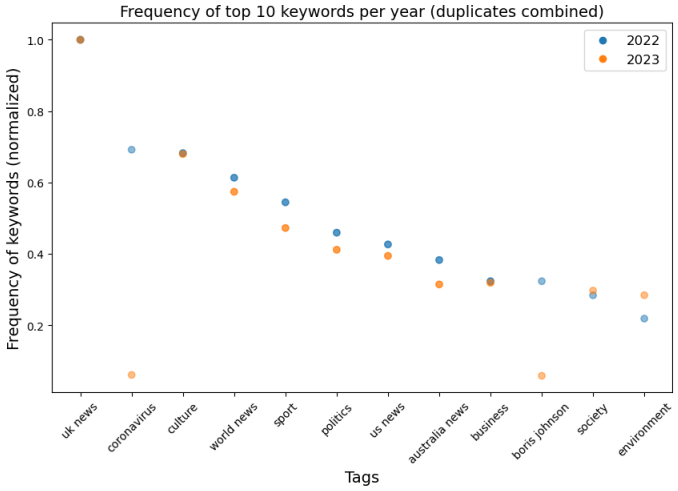

Fig. 2. Min-max normalized frequency of the top 10 tags for both years. Tags that are part of the top 10 in both years are only displayed once.

In both years ”Politics” and ”Entertainment and Culture” were the categories with the most articles published, with a pretty clear distinction to all other categories. On the other hand of the spectrum ”Science and Technology” is the category with the least amount of articles published in both years by a big margin, as the category with the second fewest articles published has more than twice the number of articles.

When we look at publishing patterns for both news outlets, they seem to follow a similar structure. There is a clearly visible dip in the amount of articles being published over the weekend. This is visualized in Fig. 3.

Investigating the relationship between the frequency of reports in a category and the corresponding public interest in that topic was done by creating a score column, this process is detailed in the *Pre-Processing* subsection of the report. The visualizations of the data for 2022 and 2023 is displayed in Fig. 4.

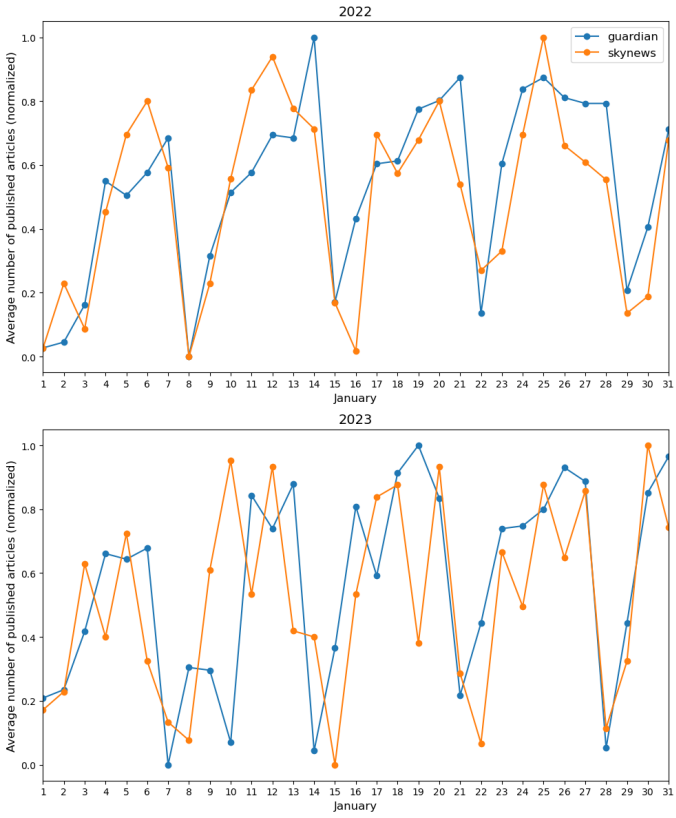

Fig. 3. Average number of articles per day, normalized for both news sites.

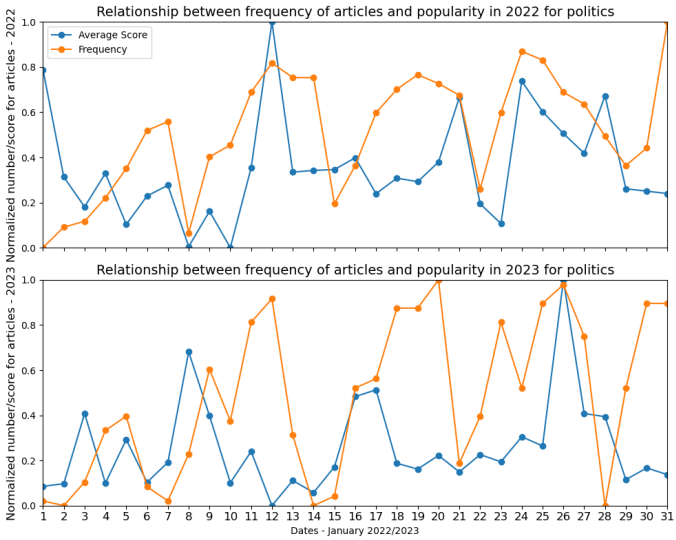

Fig. 4. Visualization of the frequency of published articles and the corresponding Google Trends popularity rating for January 2022 and 2023 (min-max normalized).

The visualizations in Fig. 4 depict the relationship between the amount of articles posted which are labeled with tags related to politics and the average score based on the popularity of the tags related to the same category for each day. As can be seen in the visualizations, there does exist a basis for stating that there exists at least a low correlation between the news coverage on a certain topic, and the corresponding increase in general interest for that topic. The figures show that generally if news coverage increases, there exists an incline in the the public interest in the following period. Similarly, a decrease in news coverage on a specific topic is generally followed by a decrease in public interest. Some uncorrelated events also occur in the figures. Although it would be hard do accurately specify the reason for this, it is expected to some degree due to there being potential influential sources outside of the data present in this study.

Visualizations of the same relationship for the categories ”Entertainment and culture”, ”Health”, ”Sports”, ”Business and economics” and ”Science and technology” can be found under VII-C.

TABLE III
DELAYED CORRELATION ANALYSIS 2022

Category	Correlation		
	Lag 0	Lag 1	Lag 2
entertainment and culture	0.208	0.349	0.189
sports	−0.324	0.362	0.211
health	0.412	0.226	−0.011
environment	−0.059	−0.076	0.111
business and economy	0.369	0.030	−0.346
science and technology	0.061	0.438	−0.065
politics	0.299	0.467	0.260

TABLE IV
DELAYED CORRELATION ANALYSIS 2023

Category	Correlation		
	Lag 0	Lag 1	Lag 2
entertainment and culture	−0.141	0.074	0.004
sports	−0.409	0.115	0.245
health	0.237	0.175	−0.245
environment	0.236	−0.149	−0.026
business and economy	0.323	0.023	−0.429
science and technology	0.164	−0.129	−0.244
politics	0.182	0.018	−0.167

Table III and Table IV, describes the delayed pearson correlation coefficient for the frequency of articles published and the interest of the population. The columns ”Lag 0”, ”Lag 1” and ”Lag 2” shows the correlation for the same day, one day and two days after the articles were published. For most article categories we see an increase in the correlation the day of or the day after the article frequency in that category increased. For the second day we generally see that the correlation decreases compared to previous days. The data suggests that articles influence the population search results in a relatively short time period following the publication. Additionally the tables confirm that there exists a correlation between published articles and public interest.

V. Discussion

A. Data collection

The API provided by *The Guardian* for accessing their content, including articles, is very comfortable and easy to use. By specifying the timeframe and types of content that should be extracted it is straightforward to get relevant data for our analysis. The rate limit on API calls that can be made per day with a free developer key did not prove to be an issue for the scale of our project, but could be problematic for other use cases.

Since we only focused on a relatively short timeframe of one month for our data collection, our results are somewhat limited as that they might be subject to seasonality or periodicity. To get a more complete insight into how reporting in the media looks and changes over time, a longer timeframe for observation would be interesting, possibly a calendar year to cover all possible occurrences of seasonality.

Our final dataset for the article data is made up of 11.978 articles from *The Guardian* and 3.422 articles from *Sky News*. This clearly shows that the analysis will be biased towards *The Guardian*, since the amount of data is much greater.

B. Data Annotation

Our annotation process has some shortcomings that we tried to reduce, but we could not alleviate them completely. For the manual annotation, the number of individuals performing the annotation is very low with only two individuals, who both have similar demographic backgrounds. This certainly leaves room for bias in the way articles are grouped into categories. Ideally the number of manual annotators would have been higher, as well as them having a more diverse demographic background.

When looking at the automatic annotation, using GPT-3.5-Turbo proved to be decently accurate, as discussed in section III-B2. Regardless, there is still some level of inaccuracy in using such a model, since articles can be reasonably classified into two different categories at the same time, which we tried to control for by showing the accuracy of annotations where GPT-3.5-Turbo agrees with both manual annotators.

Another aspect to consider when using GPT-3.5-Turbo is the way the model was trained. A lot of large language models are trained with datasets consisting of text that was acquired by using web crawlers. These datasets are intentionally very large, but still they do not guarantee diversity in their content, since usage of the internet is more biased towards males from western countries. This means that certain biases that this demographic group carries might be included in the language model [16].

Currently the categorization of the tags related to an article is based on six relatively broad categories, generally these categories depict the different areas of information quite well. However, we noticed that the ”Entertainment and Culture” category contains articles that might be better categorized into a ”society” category.

For the annotation we focused only on the headline and tags of an article. One option to increase the accuracy of the categorisation would be by also taking in to account the whole text of an article. This would provide more insights into the content of an article and what it is about. We decided to only use the headline and tags due to API limitations, because we only had a limited amount of credits which decrease by the number of used tokens, so incorporating the complete article text into each request would have used up too many tokens. This could have been circumvented by paying for a higher access tier to the GPT-3.5-Turbo model.

Using the complete article text also would have enabled us to include all articles in our analysis, even the ones that do not have any tags associated with them, possibly enabling a more complete analysis.

C. Analysis

Considering the analysis there exist cases where the results are not entirely accurate. The data present in this study is from a subset of the news corporations available in the UK, while the results from the *pytrends* are from the whole UK population. Therefore, there will be some degree of outside influence present in the data, both from other news outlets and sources unrelated to news corporations. While the insights gained from the analysis still reflect some truth, it is important to keep in mind that it does not reflect the entire image.

The same argument can be made for other countries than the UK. Although it is probably safe to assume that there exists some degree of likeness in different countries, there may also be large differences. Therefore, the insights reported in this study should not be directly applied as universal truth in other areas. There might be a cultural bias toward certain categories of topics in the UK, that does not exist in that form in other countries, which would make our analysis inapplicable for that country.

Another potential limitation of the accuracy of the analysis performed is the accuracy of the automatic annotation. As stated earlier, the annotation is not absolutely accurate, which could introduce some degree of error.

In the analysis of the most represented topics of 2022 and 2023 we see a comparatively small amount of science and technology articles published for both years. One reason for this might be that science and technology articles are heavily represented by publishers that focus specifically on these types of articles, and due to the technicality of these articles they might not be interesting or understandable to

the general population. Therefore the science and technology sections may be misrepresented in the mainstream media due to the lack of general interest, while collectively these articles could account for a more substantial part of what interests a population as a whole. This potential misrepresentation could also affect the results visualized in fig. 3 and fig. 4.

The potential misrepresentation discussed for science and technology might also be present in other categories. The fact that there are publishers that are more heavily focused on specific subjects might entail that people generally look at different sources for different categories of information. Meaning that people might read certain publications from specific providers to get information on business and economy and use another provider to get information on health. The news providers chosen in this study generally have a relatively diverse selection of material from different categories, which was one of the reasons why we chose them to try and counteract this issue. However, it is important to keep in mind that this might still have an affect the accuracy of the results shown in the analysis.

D. Research questions

In the analysis we stated that there can be seen an increase in public interest following an increase in the media coverage for a specific category. while this is true, some parts of the visualizations in fig. 4 could also be interpreted as the media increasing it's coverage on a particular subject as a product of public interest. The answer to the research questions concerning the influence of the media on public opinion if the data is interpreted this way may be that there is no clear answer. Therefore the arguments for that the media influences the public interest and that the public interest influences the media may both be true in some sense.

VI. Conclusion

A. Summary

In this study we have investigated the relationship between two of the largest news providers in the UK and the corresponding trending searches for the population of the UK. We performed both an explanatory analysis of the data and an analysis of the connection between media and public interest. The results gained through the analysis showed us that there is a connection between media coverage and public interest, although for some cases it is questionable whether the media influences the public or the public influences the media.

B. Outlook

To increase the accuracy of the automatic annotation the whole text of an article could be used for the annotation. Applying some further Natural Language Processing procedures in order to perform the annotation based on the text could account for some errors made by the current annotation procedure. Furthermore, utilizing the text of an article to annotate could remove mismatches due to some ambiguity where the article may touch upon several categories, but realistically only be focused on one category in particular.

The article text could also be used to compare the reporting of different news sites to one another, in terms of the tone they are using when reporting on the same issues or specific news stories. Furthermore, this could be used to determine the political alignment of news sites, or if they are biased in their reporting about certain topics or categories of topics.

Another possible use case for our dataset could be an analysis on how different authors present the same topics in articles that they authored, in a similar vein as outlined above for news sites. By grouping authors that often collaborate on articles, it would also be possible to get an insight into the way specific editorial offices at news sites work and collaborate, identifying different departments and specialisations.

References

[1] Ofcom, (2023). News consumption in the UK: 2023. Ofcom.org.uk. https://www.ofcom.org.uk/__data/assets/pdf_file/0024/264651/news-consumption-2023.pdf

[2] Aisha Majid, (2023, November 26). Top 50 UK news websites: ITV and Mail Online see double-digit growth in October. https://pressgazette.co.uk/media-audience-and-business-data/media-metrics/most-popular-websites-news-uk-monthly-2/

[3] Internet Archive, (n.d.). Internet Archive Wayback Machine. https://archive.org/web/https://archive.org/web/

[4] The Guardian, (2023). https://open-platform.theguardian.com/access/

[5] The Guardian, (2023). https://open-platform.theguardian.com/documentation/search

[6] codelucas, (2020). Newspaper3k: Article scraping & curation. GitHub. https://github.com/codelucas/newspaper

[7] Leonard Richardson, (2023). Beautiful Soup. https://www.crummy.com/software/BeautifulSoup/

[8] Python Software Foundation, (2023). Requests: A simple, yet elegant, HTTP library. GitHub. https://github.com/psf/requests

[9] lxml, (2023). Library for processing XML and HTML in the Python language. GitHub. https://github.com/lxml/lxml

[10] SeleniumHQ, (2023). Selenium. GitHub. https://github.com/SeleniumHQ/selenium

[11] GeneralMills, (2023). pytrends. GitHub. https://github.com/GeneralMills/pytrends

[12] sundios, (2023). pytrends issue 561. GitHub. https://github.com/GeneralMills/pytrends/issues/561

[13] Hubaib, (2023, March 15). Pytends api throwing 429 error even if the request was made very first time. Stack Overflow. https://stackoverflow.com/questions/75744524/pytends-api-throwing-429-error-even-if-the-request-was-made-very-first-time?

[14] Akanksha, (2020, May 28). Compare more than 5 keywords in Google Trends Search using pytrends. Medium. https://medium.com/analytics-vidhya/compare-more-than-5-keywords-in-google-trends-search-using-pytrends-3462d6b5ad62

[15] OpenAI, (2023). Introducing ChatGPT and-whisper API. OpenAI. https://openai.com/blog/introducing-chatgpt-and-whisper-apis

[16] Bender et al. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?, FAccT '21, March 3–10, 2021, Virtual Event, Canada

VII. Appendix

A. Prompt for batch annotation

You are an excellent annotator. You will be provided with a batch of items. Each item consists of a title and one or multiple tags that both belong to one news article. The items start with *itemStart* and end with *itemEnd*. Your task is to

classify the topic of a news article by looking at its title and tags in either Category-1: politics, Category-2: business and economy, Category-3: environment, Category-4: sports, Category-5: entertainment and culture, Category-6: science and technology or Category-7: health. Return for each article only the number of the category in a python list where the result matches the order of the input.

B. Prompt for single annotation

You are an excellent annotator. You will be provided with a title in between *titleStart* and *titleEnd* and one or multiple tags in between *tagStart* and *tagEnd*. Both the title and tags belong to a news article. Your task is to classify to which of the following categories the article most likely belongs to. However, you are only allowed to pick categories from the following list: 'Category-1: politics', 'Category-2: business and economy', 'Category-3: environment', 'Category-4: sports', 'Category-5: entertainment and culture', 'Category-6: science and technology' or 'Category-7: health'. Return the number of only one of the provided categories.

C. extra visualizations

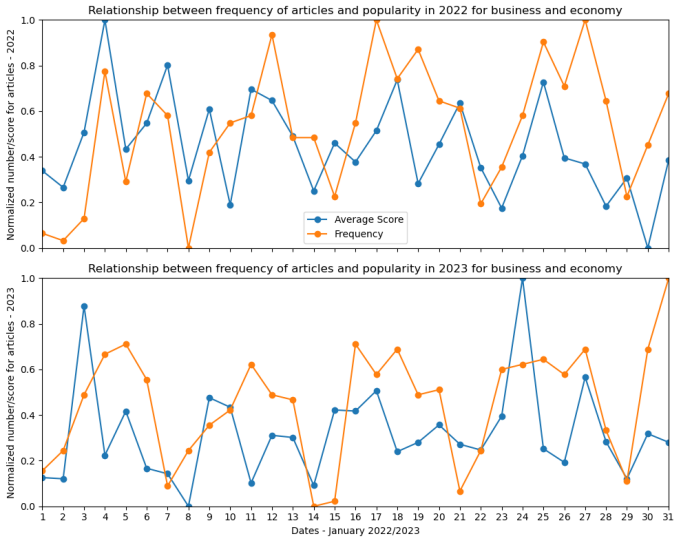

Fig. 5. Visualization of the frequency of published articles and the corresponding Google Trends popularity rating for January 2022 and 2023 (min-max normalized).

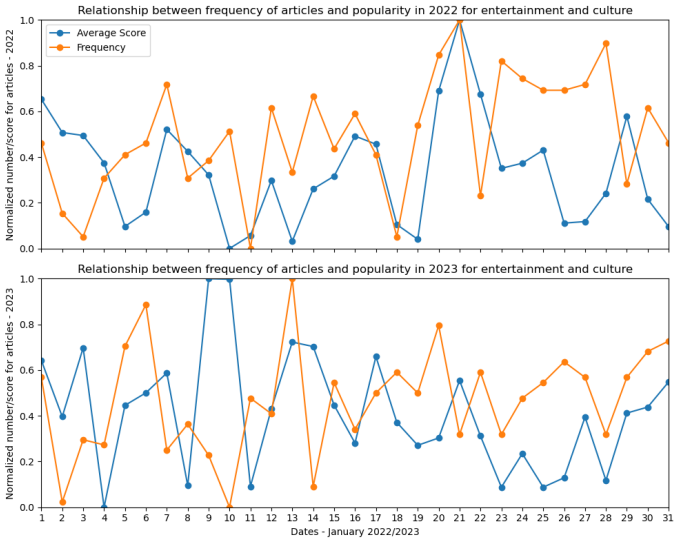

Fig. 6. Visualization of the frequency of published articles and the corresponding Google Trends popularity rating for January 2022 and 2023 (min-max normalized).

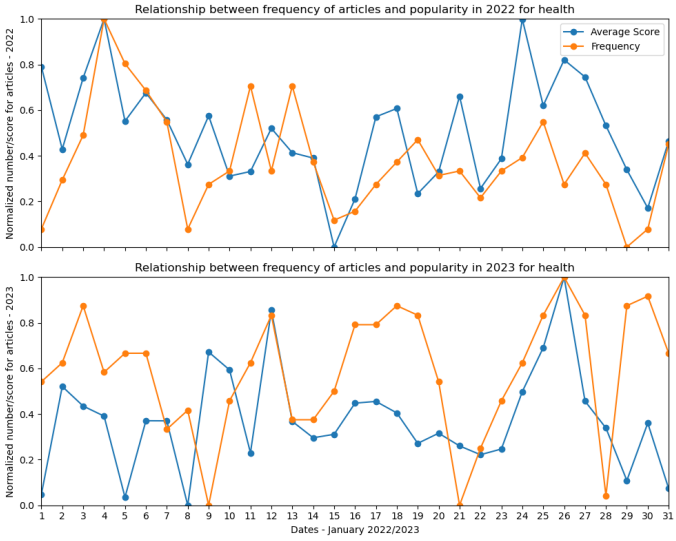

Fig. 7. Visualization of the frequency of published articles and the corresponding Google Trends popularity rating for January 2022 and 2023 (min-max normalized).

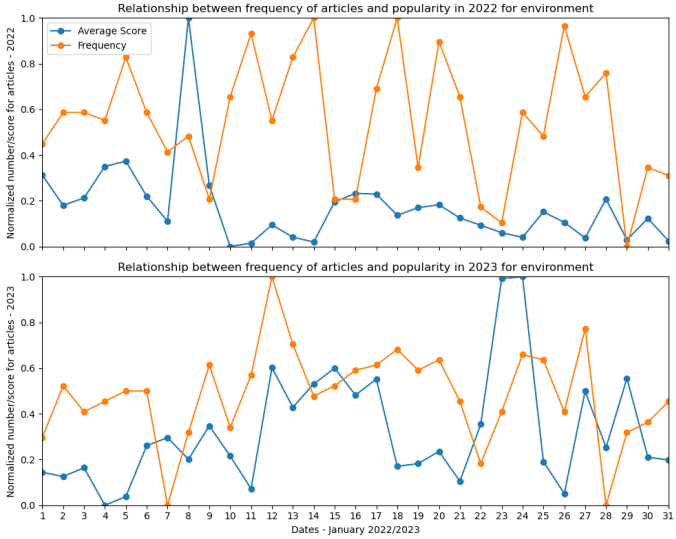

Fig. 8. Visualization of the frequency of published articles and the corresponding Google Trends popularity rating for January 2022 and 2023 (min-max normalized).

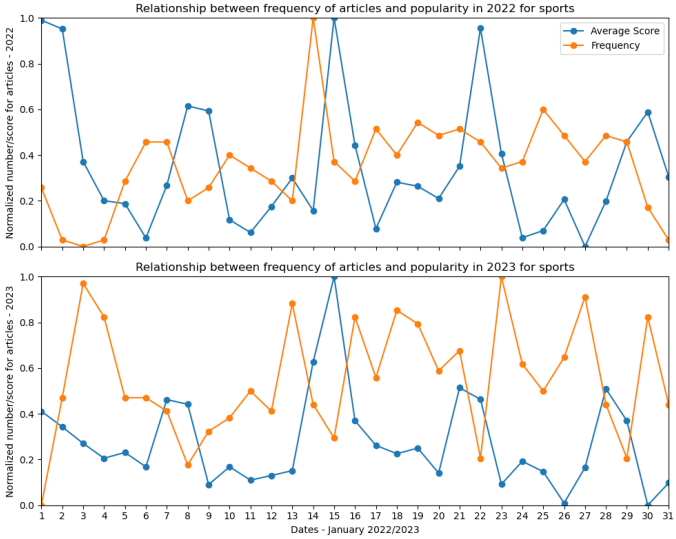

Fig. 10. Visualization of the frequency of published articles and the corresponding Google Trends popularity rating for January 2022 and 2023 (min-max normalized).

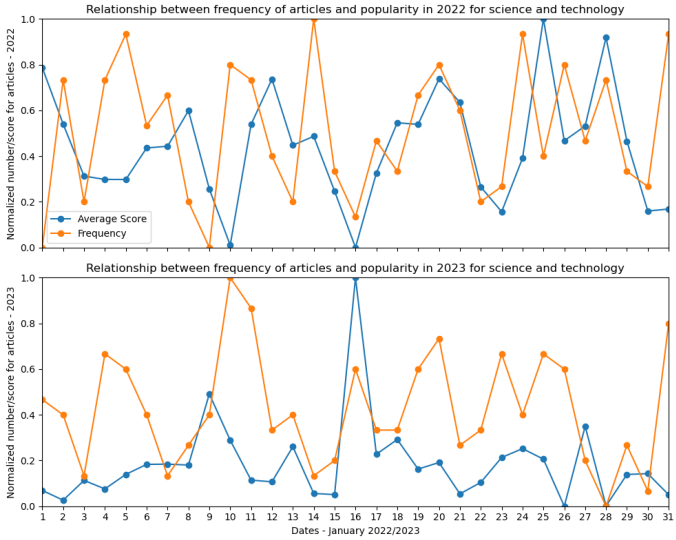

Fig. 9. Visualization of the frequency of published articles and the corresponding Google Trends popularity rating for January 2022 and 2023 (min-max normalized).